UNIVERSITÉ DE MONTRÉAL

CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC, H3C 3J7

ROUTING AND SCHEDULING BY BRANCH-AND-BOUND ON TIME WINDOWS

by

Jacques Desrosiers, François Soumis
Martin Desrochers and Michel Sauvé

Présenté au "Third International Workshop onf Transit Vehicle and Crew Scheduling", Centre de recherche sur les transports, Université de Montréal, 26-30 juin 1983.

A paraître dans Computer Scheduling of Public Transport, Volume II, North-Holland Publishing Company (1984).

Université de Montréal
Centre de recherche sur les transports - Publication #278
Ecole des Hautes Etudes Commerciales -
Cahiers du GERAD #G-83-16
Août 1983

Dépôt légal - 1er trimestre 1984

ABSTRACT

Consider a set of trips where each trip is specified a priori by a place of origin, a destination, a duration, a cost and a time interval within which the trip must begin. The trips may include visits to one or more specific points. Our problem is to determine the number of vehicles required together with their routes and schedules, so that each trip begins within his given interval, while the fixed costs related to the number of vehicles, and the travel costs between trips are minimized. The problem is a generalization of the m-travelling salesman problem.

We compare numerical results for two algorithms :

1) Adaptation of the Carpaneto-Toth algorithm for the asymmetric travelling salesman problem : solution of network problems by relaxing scheduling constraints, and branch-and-bound on flow variables.

2) Solution of network problems by relaxing scheduling constraints and branch-and-bound based on dividing the time windows.

- - - - - - - - - - - - - - -

1. INTRODUCTION

For a transportation company, decisions concerning vehicle routing and scheduling are both important and difficult. They are important as they affect the cost of services offered by determining the number of vehicles used, empty running time, waisting time and productive running time. They are difficult because of the large number of possibilities resulting from the large number of interrelated variables to be considered both in space on the network, and in time relative to scheduling requirements.

1.1 Terminology

We first define the terms "trip", "intertrip arc" and "route" which are used throughout this text. A trip is a productive journey which must be carried out by a vehicle. The trip i is characterized by a place of origin, a destination, a duration, a cost and a time interval $[a_i, b_i]$ during which the trip must begin. An intertrip arc is an unproductive (i.e. empty) journey carried out by a vehicle. The intertrip arc (i,j) goes from the end of trip i to the beginning of trip j. Its duration t_{ij} and its cost c_{ij} may include respectively the duration anc cost of the trip i. A route is a sequence of trips and intertrip arcs carried out by the same vehicle.

1.2 Stages in the planning process

Vehicle use planning is often done in two stages as follows :
1. Determine the trips to be carried out based on customer demand.
2. Construct an optimal set of routes to deal with all the trips.

The first step deals with the demand without explicitly considering the entire network. The demand is partitioned and trips are defined to serve each part. The objectives and methods used at this stage vary. Trips may be constructed to satisfy a priori certain local constraints which would be difficult to handle at the route construction stage: e.g. constraints on the compatibility of items transported, on the order of loading and unloading, and on vehicle capacity. Trips can be constructed to aggregate demand by grouping clients who can best be served in a given order by the same vehicle. The level of aggregation should be chosen taking into consideration the size of the problem which can be handled in step 2 and the degree of optimality desired. If no aggregation is done, a trip is defined for each client and the entire problem is handled in the second step.

The second step determines how the transportation service is organized for the whole network. This is the problem handled by the medhod proposed in this paper.

1.3 Mathematical formulation

In order to construct a set of optimal routes, the intertrip arcs used and the starting times for each trip must be determined. These variables must be fixed so as to minimize the number of vehicles and travel costs while respecting the network and scheduling constraints.

Let P be the set of trips and I the set of intertrip arcs considered. In order for an intertrip arc to be considered, it must be possible to carry out trip j after trip i while respecting the time interval constraints. An arc $(i,j)\in I$ is defined as the movement of a vehicle from the end of trip i to the beginning of trip j. An arc $(i,j)\in I$ can be defined if the following relation holds :

$$a_i + b_{ij} \leq b_j \qquad i,j\in P \qquad (1)$$

The set I is generally restricted by other a priori constraints on empty running : for example, in air transportation, a plane completing a flight and landing in a given airport can only be assigned to flights departing from this same airport.

We examine the problem with a single depot which each vehicle leaves only once on a route. The network used by the vehicles is defined by the set of nodes $N = P \cup \{s\} \cup \{t\}$, and the set or directed arcs $A = I \cup \{s\}xP \cup Px\{t\}$, where s and t represent the entrance to and exit from the depot respectively.

In the problem formulation, two types of variables are used : flow variables x_{ij}, $(i,j)\in A$ taking the value 1 when arc (i,j) is used by a vehicle, and continuous time variables t_i, $i\in P$ associated with the starting time of each trip. The variable travel costs are associated with the flow variables corresponding to the intertrip arcs, and the fixed cost C per vehicle is associated with the flow variables corresponding to the arcs of the set $\{s\}xP$.

Optimal routes respecting the scheduling constraints are the solution of the following problem :

$$\text{Min } C^* \sum_{j\in P} x_{ij} + \sum_{(i,j)\in I} c_{ij}\, x_{ij} \qquad (2)$$

$$\text{subject to} \quad \sum_{j\in N} x_{ij} = \sum_{j\in N} x_{ji} = 1 \qquad i\in P \qquad (3)$$

$$x_{ij} \geq 0 \qquad (i,j)\in A \qquad (4)$$

$$x_{ij} > 0 \Rightarrow t_i + t_{ij} \leq t_j \qquad (i,j)\in I \qquad (5)$$

$$a_i \leq t_i \leq b_i \qquad i\in P \qquad (6)$$

Relations (2), (3) and (4) form a simple routing problem without scheduling constraints. This is a minimum cost flow problem having an integer solution. Relations (5) describe the compatibility requirements between the routes and the schedules, while constraints (6) establish the time interval within which trips must begin.

For the vehicle routing problem with scheduling constraints defined by (2) - (6), it is easily shown [8] that there exists an integer optimal solution for the flow variables. However, this solution cannot be obtained

directly by linear programming as the relations (5) are not linear constraints. These relations can however be written in a linear form :

$$t_i + t_{ij} - t_j \leq (1 - x_{ij})\, M_{ij} \qquad (i,j)\in I \qquad (7)$$

where $M_{ij} \geq b_i + t_{ij} - a_j$. This formulation is equivalent to (5) only if we require that x_{ij} be integer; without the integrality constraint, the optimal solution may be fractional and it will depend on the values of the constraints M_{ij}. Finally, the classic travelling salesman problem is a special case of this formulation when

$$(\forall\, i\in P)\ [a_i, b_i] = [0, |P|-1] \text{ and } (\forall\, (i,j)\in I)\ t_{ij} = 1;$$

in this case, we may set $M_{ij} = |P|$ and constraints (7) become the subtour elimination constraints proposed by Miller, Tucker and Zemlin [13] :

$$t_i - t_j + |P|\, x_{ij} \leq |P| - 1 \qquad (i,j)\in I\ . \qquad (8)$$

In this special case, our problem reduces to a travelling salesman problem, while in the general case, it is an m-salesman problem with time windows where the scheduling constraints ensure the elimination of subtours. For the routing problem with scheduling constraints, several constraints of type (6) and (7) are usually active at the optimum. This gives an indication of the difficulty of the problem knowing how difficult the travelling salesman problem can be even though few subtour elimination constraints are active at the optimum.

2. THE LITERATURE

This route construction problem has been encountered by the authors in several contexts : e.g. school busing [6], urban transit [16], passenger trains [15], airline fleets [17]. The same problem structure also occurs in the assignment to m machines of n tasks which must be carried out within a given time interval. Bratley, Florian and Robillard [3] and Gertsbakh and Stern [11] have examined this context.

2.1 Heuristic methods

One possible approximate solution method is to replace each variable t_i by a set of binary variables associated with each decision to begin a trip i or not at a number of discrete points within the time interval. This method has been used by Levin [12] for air transportation, and by Swersey and Ballard [18] for school transportation; they frequently obtained an optimal integer solution using the simplex algorithm without integrality constraints. Note, however, that they restricted the objective function to include only the number of vehicles, after noting that it was more difficult to obtain integrality when travel costs were also included. Other methods proposed by Bodin et al. [2] include travel costs but these require some additional approximations.

2.2 Exact methods

Orloff [14] presented a first attempt at an optimal approach to the problem. He proposed the reformulation of the problem as a set partitioning

problem by explicitly enumerating all routes satisfying the scheduling constraints. However, Orloff notes that the number of variables is very large even for a medium-sized problem and this approach cannot therefore be successfully applied to real problems.

Branch-and-bound methods have been most commonly used to solve this type of problem optimally. A relaxed problem is solved and branching is carried out to eliminate infeasible solutions. Several relaxations have been proposed; they present different compromises between solution cost and quality of approximation. For the asymmetric travelling salesman problem, Carpaneto and Toth [5] relaxed subtour elimination constraints to obtain an easily solvable assignment problem, and they improved on Bellmore and Malone's [1] branching on the flow variables to break the subtours. This method has been adapted to our more general problem; when constraints (6) and (7) are very restrictive the gap between the assignment problem and the original problem is considerable and the branch-and-bound tree to be explored may become very large. To reduce this gap and the size of the tree, Fisher [9] has proposed a Lagrangian relaxation while Christofides, Mingozzi and Toth [4] have proposed state-space relaxation which uses intermediate relaxations close to the original problem starting from shortest path or spanning tree problems.

The real number relaxation obtained by replacing (5) by its linear equivalent (7) and relaxing the integrality requirement produces a smaller gap than the previous relaxations (Geoffrion [19]). This approach was not used as the solution by Simplex of a problem with $3*|P| + |I| = O(|P|^2)$ constraints is too costly. Desrosiers, Soumis and Desrochers [8] use this relaxation but solve the linear problem by column generation (Dantzig-Wolfe) to take advantage of the network structure of the problem. The master problem is a partitioning problem with $|P|$ constraints and can be solved by Simplex. The subproblem is a shortest path problem with time interval constraints on the nodes [7]. Integrality is obtained through a very small implicit enumeration tree which is further reduced as necessary by two cuts requiring that the number of vehicles and/or the travel costs be rounded up to the next integer. An important property of this algorithm is that the time variables t_i are considered as continuous variables with no a priori discretization being necessary.

In this article, we relax constraints (6) and (7) and we examine a new branching scheme which works not by fixing the flow variables x_{ij}, but by dividing up the time intervals $[a_i, b_i]$

3. THE NETWORK RELAXATION

This solution method relaxes the scheduling constraints (5) and (6) and solves a network problem at each node of the branch-and-bound tree. The optimal solution of each network problem is integer and involves the construction of a set of routes and sometimes some subtours. The schedule t_i of each trip $i \in P$ must be determined to check whether constraints (5) and (6) are satisfied.

3.1 Determination of the scheduling component

A route of m trips is specified by a sequence of arcs $(i,j) \in A$

$$(s,i(1)),\ (i(1),i(2)),\ \ldots,\ (i(m),t)$$

whose associated x_{ij} variables all take the value 1, or equivalently by the sequence of network nodes

$$s,\ i(1),\ i(2),\ \ldots,\ i(m),t$$

where s and t represent the depot.

A subtour of n trips is specified by the arcs :

$$(i(1),\ i(2)),\ \ldots,\ (i(m),\ i(m+1))$$

or equivalently by the sequence of nodes

$$i(1),\ i(2),\ \ldots,\ i(m),\ i(m+1)$$

where nodes i(1) and i(m+1) are identical.

The first trip i(1) of a subtour is chosen so that $a_{i(1)} = \min_{1 \le r \le m} \{a_{i(r)}\}$.

As equation (5) allows waiting before the beginning of a trip, the time variables are evaluated as follows :

$$\begin{aligned} t_{i(1)} &= a_{i(1)} \\ t_{i(r)} &= \max \{a_{i(r)},\ t_{i(r-1)} + t_{i(r-1),\ i(r)}\} \quad 2 \le r \le m \end{aligned} \tag{9}$$

with an additional calculation at node i(1) of a subtour :

$$t_{i(m+1)} = t_{i(m)} + t_{i(m),\ i(1)} \tag{10}$$

When there are no subtours and the time window constraints are satisfied for all time variables, the solution of the network problem is feasible for the original problem. Otherwise branching is carried out either by fixing the flow variables on the arcs at 0 or 1, or by dividing up the time intervals for the trips. Two cases are considered :

Case 1 : For a route or a subtour of m trips, there is at least one time variable from among $t_{i(1)}$, $t_{i(2)}$, ..., $t_{i(m)}$, $t_{i(m+1)}$ which exceeds the upper limit of his time window.

Case 2 : For a subtour of m trips, all the time variables $t_{i(1)}$, $t_{i(2)}$,..., $t_{i(m)}$, and $t_{i(m+1)}$ satisfy the time window constraints.

For a subtour in Case 2 containing m trips, we have $t_{i(r)} \in [a_{i(r)}, b_{i(r)}]$, $1 \cdot r \le m$ and also $t_{i(m+1)} \le b_{i(1)}$. The whole set of arcs or nodes of this subtour must be considered in developing a branching rule. On the other hand, in Case 1, the analysis can be restricted to part of the route or subtour considered.

Let $i(\gamma)$ be the first trip for which the time variable $t_{i(\gamma)}$ exceeds the upper bound $b_{i(\gamma)}$ and define the excess time $\delta > 0$ by

$$\delta = t_{i(\gamma)} - b_{i(\gamma)} \tag{11}$$

Note that $\gamma \leq m$ for a route, while $\gamma \leq m+1$ for a subtour. Let $i(\alpha)$ be the last trip before $i(\gamma)$ for which $t_{i(\alpha)} = a_{i(\alpha)}$:

$$\alpha \quad \max \{1 \leq r \leq \gamma - 2 \mid t_{i(\alpha)} = a_{i(\alpha)}\} \tag{12}$$

The fact that α must be less than or equal to γ-2 implies that an illegal route contains at least three trips (an arc between two trips always satisfies relation (1)) and that a subtour always contains at least two distinct trips. The section of the path to be analyzed between $i(\alpha)$ and $i(\gamma)$ can be further restricted by calculating the latest time $t^{-1}_{i(r)}$ which would have to have been assigned to trip $i(r)$ preceding trip $i(\gamma)$ so that it could begin at time $t_{i(\gamma)} = b_{i(\gamma)}$. Letting $t^{-1}_{i(\gamma)} = b_{i(\gamma)}$, the values $t^{-1}_{i(r)}$, $\alpha \leq r \leq \gamma-1$ are calculated according to the following recurrence formula :

$$t^{-1}_{i(r)} = \min \{b_{i(r)}, t^{-1}_{i(r+1)} - t_{i(r+1),i(r)}\} \quad \alpha \leq r \leq \gamma-1 \tag{13}$$

or more simply, following from the definitions of α and δ :

$$\begin{aligned} t^{-1}_{i(r)} &= t^{-1}_{i(r+1)} - t_{i(r+1),i(r)} && \alpha \leq r \leq \gamma-1 \\ &= t_{i(r)} - \delta && \alpha \leq r \leq \gamma-1 \end{aligned} \tag{14}$$

Obviously, $t^{-1}_{i(\alpha)} = a_{i(\alpha)} - \delta < a_{i(\alpha)}$ does not satisfy the time window constraint; then let $i(\beta)$ be the last trip before $i(\beta)$ for which $t^{-1}_{i(\beta)}$ does not satisfy the time interval constraint :

$$\beta = \max \{\alpha \leq r \leq \gamma - 2 \mid t^{-1}_{i(r)} < a_{i(r)}\} \tag{15}$$

We therefore have $1 \leq \alpha \leq \beta \leq \gamma - 2$, and the section of the path between nodes $i(\beta)$ and $i(\gamma)$ is infeasible. Figure 1 illustrates an example with seven trips where $\alpha = 2$, $\beta = 3$ and $\gamma = 7$, which restricts the development of branching procedures to the section included between β and γ.

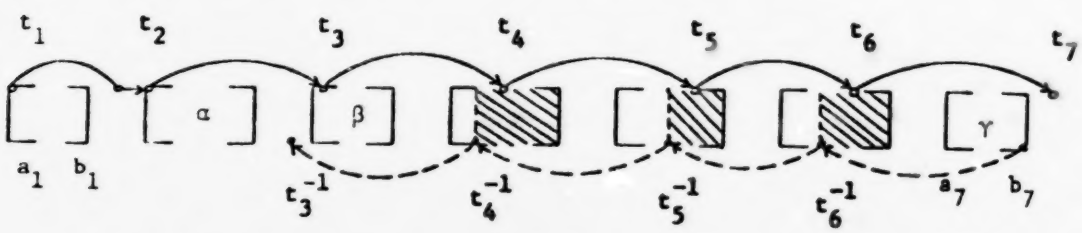


Figure 1
Description of time variables on a route which violates the time interval constraints

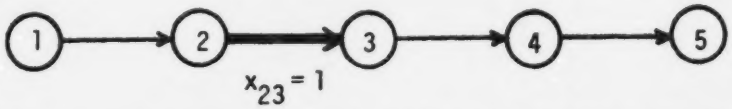
3.2 Branching on the flow variables

The branching on the flow variables at a node of the branch-and-bound tree follows the method introduced by Bellmore and Malone [1] and improved by Carpaneto and Toth [5] for the asymmetric travelling salesman problem. This branching is applied in the same way for a subtour and for a section

of route, with no real distinction between Cases 1 and 2.

The method first fixes at 0 or at 1, the k free x_{ij} variables associated with the arcs making up the subtour or section of route considered. Among the 2^k ways of fixing these binary variables, the current infeasible solution (i.e. that in which all free variables equal 1) must be eliminated. The other 2^k-1 possibilities are grouped into k branches containing respectively 1,2, 4, ... 2^{k-1} possibilities, and each branch explicitly eliminates a free arc of the subtour or illegal route considered. The first branch fixes the first k-1 free variables at 1 and the last at 0. The second branch fixes the first k-2 variables at 1 and the next at 0; the final variable therefore remains free. And so on, until the k^{th} branch where the first free variable is fixed at 0, leaving the other variables free.

The enumeration process is illustrated in figure 2. In addition to explicitly eliminating a free arc at each branch, three additional sets of arcs can also be eliminated, with the last set being specific to the routing problem with scheduling constraints. Consider a directed path from i(1) to i(h) where all variables on the path have been fixed at 1, either at this level in the tree or at a lower level, i.e.



Subtour or route section violating scheduling constraints with $X_{23} = 1$ fixed at a lower level in the tree (3 free arcs)

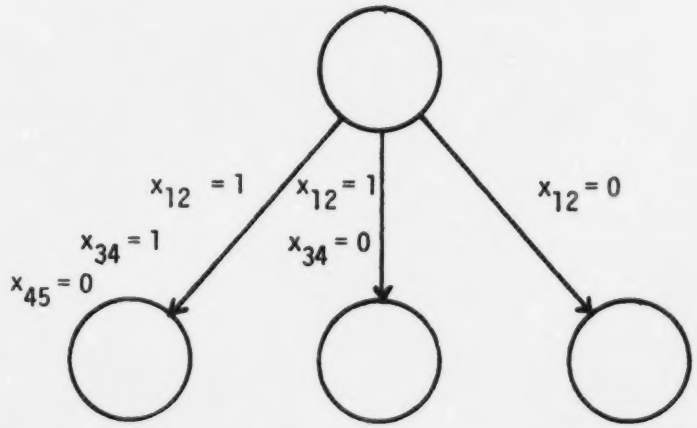


Figure 2
Flow variable branching tree

$$x_{i(r),i(r+1)} = 1 \qquad r = 1,2,\ldots,h-1$$

A) Elimination of incident arcs

$$\text{for } r = 1,2,\ldots,h-1 \qquad x_{i(r),j} = 0 \quad \forall\, j \neq i(r)$$

$$r = 2,3,\ldots,h \qquad x_{j,i(r)} = 0 \quad \forall\, j \neq i(r)$$

B) Elimination of the subtour $i(1),\ldots,i(h),i(1)$:

$$x_{i(h),i(1)} = 0$$

C) Elimination of predecessors of $i(1)$ and successors of $i(h)$:

The time intervals at $i(1)$ and $i(h)$ are reduced and then relation (1) is applied. As in equations (9) and (13), let

$$t_{i(1)} = a_{i(1)}$$

$$t_{i(r)} = \max \{a_{i(r)}, t_{i(r-1)} + t_{i(r-1),i(r)}\} \qquad 2 \leq r \leq h \qquad (16)$$

$$t^{-1}_{i(h)} = b_{i(h)}$$

$$t^{-1}_{i(r)} = \min \{b_{i(r)}, t^{-1}_{i(r+1)} - t_{i(r),i(r+1)}\} \qquad 1 \leq r \leq h-1 \qquad (17)$$

The values t_i and t_i^{-1} respectively allow the adjustment of the lower and upper bounds of the intervals :

$$[a'_{i(r)}, b'_{i(r)}] := [t_{i(r)}, t^{-1}_{i(r)}] \qquad r = 1,2,\ldots,h \qquad (18)$$

3.3 Branching on the time intervals

The division of the time intervals forms a partition similar to that produced by the binary flow variables; however none of the flow variables are fixed at one. Each branch of the branch-and-bound tree implicitly suppresses an arc of the route or subtour considered so that this route or subtour becomes infeasible for the new network problem. Cases 1 and 2 must however be handled differently.

3.3.1 Branching to satisfy time window constraints

Consider a section of route, or subtour as in Case 1 made up of trips $i(\beta),i(\beta+1),\ldots,i(\gamma)$. Branching is done by dividing into two parts the time intervals for trips $i(\beta+1)$ to $i(\gamma-1)$ leaving the intervals for trips $i(\beta)$ and $i(\gamma)$ intact. The division takes place at the values

$$t^{-1}_{i(r)} = t_{i(r)} - \delta \qquad r = \beta+1,\beta+2,\ldots,\gamma-1 \qquad (19)$$

defined in equation (14). The partition divides the time intervals $[a_i,b_i]$ into a left part L_i and a right part R_i; supposing, without loss of generality, that the values t_{ij} are integer, the new intervals are given by :

$$L_{i(r)} = [a_{i(r)},\ t^{-1}_{i(r)}]$$
$$R_{i(r)} = [t^{-1}_{i(r)} + 1, b_{i(r)}] \qquad \beta + 1 \le r \le \gamma - 1 \qquad (20)$$

There are $\gamma-\beta-1$ intervals to be divided into two, i.e. $2^{\gamma-\beta-1}$ possibilities grouped into $\gamma-\beta$ branches containing respectively 1, 1, 2, 4, ..., $2^{\gamma-\beta-2}$ possibilities; none of these can be excluded as infeasible. The first branch reduces the intervals of trips $i(\beta+1)$, ..., $i(\gamma-1)$ to their left intervals. The second branch reduces the interval for trip $i(\beta+1)$ to its right interval and the intervals of trips $i(\beta+2)$, ..., $i(\gamma-1)$ to their left intervals. The third branch reduces the interval of trip $i(\beta+2)$ to its right interval and the intervals of trips $i(\beta+3)$, ..., $i(\gamma-1)$ to their left intervals, with the interval fro trip $i(\beta+1)$ remaining intact. The process continues until the last branch where the interval for trip $i(\gamma-1)$ is reduced to its right interval, leaving the intervals for all other trips intact. This partition is illustrated in figure 3 with $\beta=3$ and $\gamma=7$.

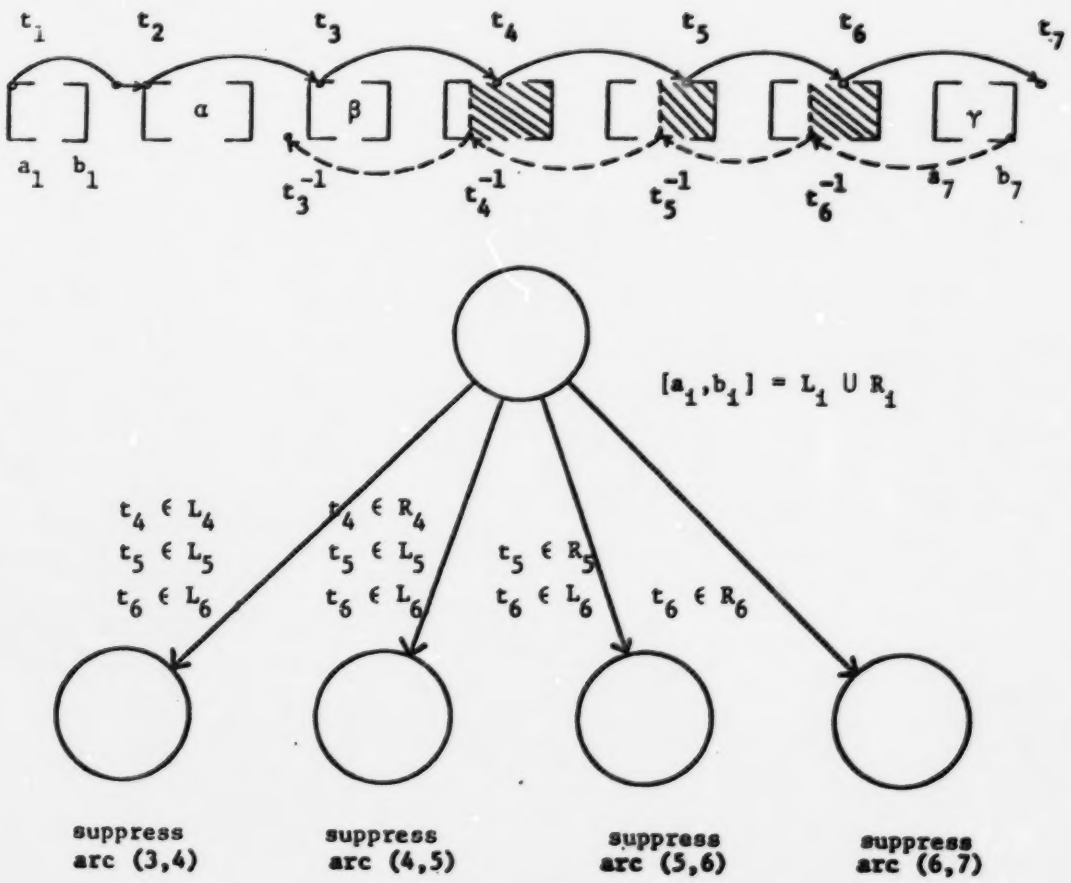


Figure 3
Branching tree to satisfy time window constraints

For a given trip, the reduction of the time interval enables incident arcs to the trip to be eliminated through relation (1). It can be shown that the first branch eliminates the arc $(i(\beta), i(\beta+1))$, the second eliminates the arc $(i(\beta+1), i(\beta+2))$ and so on until the last branch which eliminates the arc $(i(\gamma-1), i(\gamma))$. In the example in figure 3, the arcs are eliminated as follows :

$t_4 \in L_4$ eliminates arc (3,4)

$t_4 \in R_4, t_5 \in L_5$ eliminates arc (4,5)

$t_5 \in R_5, L_6 \in L_6$ eliminates arc (5,6)

$t_6 \in R_6$ eliminates arc (6,7).

The similarity between the partition on the flow variables and that on the time intervals is found by considering a section of route or a subtour of k arcs, i.e. with $\gamma-\beta=k$. In each partition, the same number of branches is obtained.

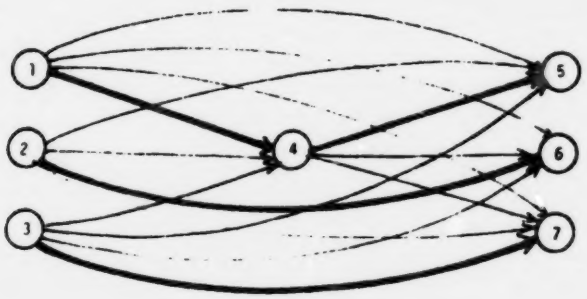
3.3.2 Branching to eliminate subtours

As defined previously, a subtour in Case 2 contrains m trips where $t_{i(r)} \in [a_{i(r)}, b_{i(r)}]$, $1 \leq r \leq m$ and $t_{i(m+1)} \leq b_{i(1)}$. This is eliminated by creating several families of branches so that in each family, this subtour no longer satisfies the time window constraints. For the first family $t_{i(1)}$ is restricted to the interval $[a_{i(1)}, a_{i(1)} + (t_{i(m+1)} - t_{i(1)})-1]$ where $t_{i(m+1)} - t_{i(1)}$ is the duration of the subtour. This subtour no longer satisfies the time interval constraint for the trip $\gamma = m+1$ and the branches described in the previous section are obtained. For $t_{i(1)} \in [a_{i(1)} + (t_{i(n+1)} - t_{i(1)}), b_{i(1)}]$, the times assigned to the trips in the subtour are recalculated and the analysis is repeated (Case 1 or Case 2). For the last family of branches (Case 1), it is possible that the first arc of the subtour may by illegal, i.e. may no longer satisfy relation (1) with the reduced interval for i(1) : in this special case, the values $\alpha = \beta = 1$, $\gamma = 2$ are obtained and the number of intervals to be divided is equal to $\gamma - \beta - 1 = 0$. The first arc is suppressed and the family only contains a single branch.

3.4 Comments on the two types of branching

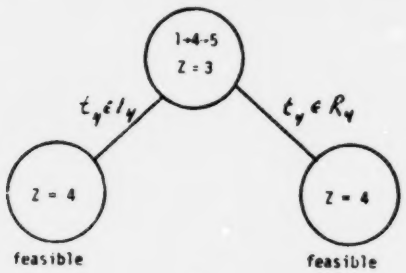
The advantage of the time interval branching method over the partition on flow variables can be clearly seen in the example in figure 4. Suppose that all routes made up of three trips fail to satisfy the scheduling constraints, and suppose also that the arrival time at trip 4 is the same whether the vehicle comes from trip 1, 2 or 3. When optimizing the number of vehicles, the network relaxation gives a solution of $z = 3$ vehicles which contains for example the illegal route $1 \rightarrow 4 \rightarrow 5$.

When branching on time intervals, the branch limiting the time variable t_4 to the left interval, simultaneously eliminates arcs (1,4), (2,4) and (3,4) and a feasible solution with four vehicles is found. For the branch in which t_4 is limited to the right interval, arcs (4,5), (4,6) and (4,7) are simultaneously eliminated and the branch-and-bound tree, containing three nodes, is completed.



(routes made up of three trips do not satisfy the scheduling constraints)

Branching on time intervals



Branching on flow variables

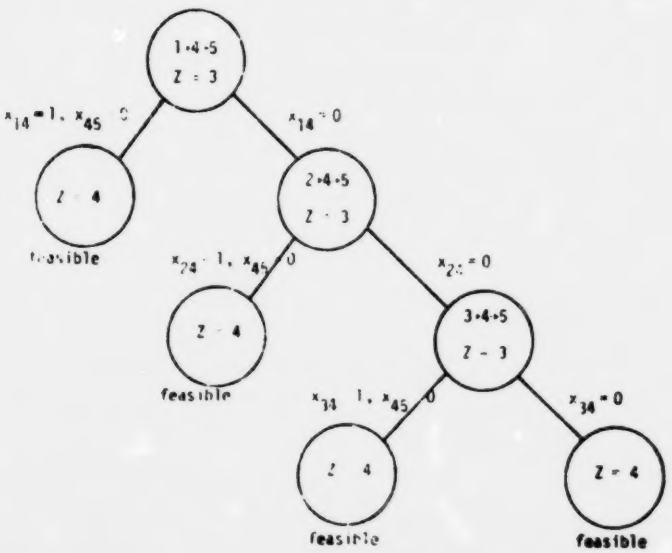


Figure 4
Comparison of branching trees

When branching on flow variables, the left branch (which fixes $x_{14} = 1$, $x_{45} = 0$) also enables arcs (4,6) and (4,7) to be eliminated through relation (1) by considering the new time interval associated with trip 4 in the route $1 \rightarrow 4$. On the other hand, for the right branch where $x_{14} = 0$, a new infeasible route is found e.g. route $2 \rightarrow 4 \rightarrow 5$. The process continues to produce a tree with seven nodes.

When elimination by bounds is used during implicit enumeration, only two nodes are explored when branching on time intervals while four are explored when branching on flow variables.

In the case of subtour satisfying the scheduling constraints (Case 2), the time interval branching creates families of branches, and the number of branches created at this level is greater than the number created by the flow variable partition. However, this type of subtour is practically inconceivable in the real applications studied (school transportation) as for a subtour of two trips, it must be possible to carry out one trip, an intertrip arc, a second trip and then an intertrip arc which brings the vehicle back to the beginning of the first trip while still respecting the time interval constraints associated with the beginning of the first trip, i.e. the following relation must be satisfied :

$$\max \left[a_{i(1)} + t_{i\,(1),i(2)}, a_{i(2)}\right] + t_{i(2),i(1)} \leq b_{i(1)} \qquad (21)$$

4. IMPLEMENTATION

4.1 Branching stragegy

The objective function (2) optimizes the number of vehicles used, and for an equal number of vehicles minimizes the sum of travel times. In our problems, the function has the following form :

$$\text{Min } Z = C * \sum_{j \in P} x_{sj} + \sum_{(i,j) \in I} t_{ij}\, x_{ij} \qquad (22)$$

where C is the fixed cost per vehicle. In our tests, the network relaxation was never feasible and the gap was often fairly large in terms of travel time and sometimes even in terms of the number of vehicles. To obtain a first feasible solution, it is necessary to descend many levels in the tree, and the number of vehicles used usually goes above the optimal number. Improvement in the objective function is therefore slow, and based mainly on travel times, although large gains can be made in terms of the number of vehicles. We therefore decided to limit the network relaxation to the optimization of the number of vehicles.

When the solution of the network problem is infeasible at a node of the branch-and-bound tree, it contains routes and/or subtours which do not satisfy the time window constraints (Case 1) and/or subtour which do satisfy these constraints (Case 2). In our application, the choice of branching method is made as follows in decreasing order of priority :

- Branching on flow variables
  1. choice of subtour (Case 1 or 2) according to estimate of bounds
  2. choice of illegal route according to estimate of bounds.

- Branching on time intervals
  1. choice of subtour (Case 1) according to estimate of bounds
  2. choice of illegal route according to estimate of bounds
  3. choice of subtour (Case 2) with maximum

$$\frac{(t_{i(m+1)} - t_{i(1)})}{(b_{i(1)} - a_{i(1)})} \quad .$$

In general, the choice of partition is first carried out on the subtours, and then on the illegal routes. However, for the time interval branching, the subtours satisfying the scheduling constraints are dealt with last as these may generate several families with several branches each; in this way, we restrict the number of nodes in the tree.

For an infeasible solution at a node in the branch-and-bound tree, let z be the value of the objective function. For each branch created, an arc (i,j) is eliminated and a lower bound $\bar{z}$ on the value of the optimal solution of the new problem is given by :

$$\bar{z} = z + \min_{j' \neq j} \{\bar{c}_{ij'}\} + \min_{i' \neq i} \{\bar{c}_{i'j}\} \quad . \tag{23}$$

If only the number of vehicles is included in the objective function, then

$$\bar{z} - z = 0 \text{ or } 1 \quad . \tag{24}$$

When the solution at a node of the tree contains several subtours in illegal routes, branching is carried out by choosing the partition which contains the smallest number of branches without increasing the bound, and in case of equality, the partition with the smallest number of branches.

Finally, nodes in the tree are explored according to the LIFO stragegy, by choosing the node having the best bound amongst those created.

4.2 Choice of the objective function

In spite of the fact that only the number of vehicles is optimized, the secondary travel time term was initially retained as it helps in generating feasible solutions.

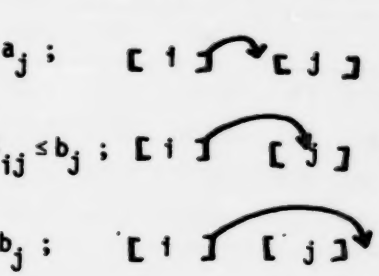
The efficiency of the network relaxation was greatly improved however by calculating at each arc $(i,j) \in I$, the probability of feasibility p_{ij} and by replacing the secondary term by the logarithm of the product of these probabilities :

$$\text{Min } Z = C * \sum_{j \in P} x_{sj} - \prod_{ij \in I} p_{ij}^{x_{ij}} \tag{25}$$

$$= C * \sum_{j \in P} x_{sj} - \sum_{ij \in I} x_{ij} \, \ell n \, p_{ij} \tag{26}$$

Supposing, with no loss of generality, that the travel times t_{ij} are integer, the time variables t_i can be considered to be discrete random variables. We then estimate the probability that arc (i,j) is feasible (in the worst case) be

$$p_{ij} = \begin{cases} 1 & \text{if } b_i + t_{ij} \le a_j \text{ ;} \\ \dfrac{b_j - (b_i + t_{ij}) + 1}{b_j - a_j + 1} & \text{if } a_j < b_i + t_{ij} \le b_j \text{ ;} \\ \dfrac{1}{(b_i - (b_j - t_{ij}) + 1)(b_j - a_j + 1)} & \text{if } b_i + t_{ij} > b_j \text{ ;} \end{cases} \qquad (27)$$



In fact p_{ij} estimates the proportion of pairs (t_i+t_{ij}, t_j) satisfying the relation $t_i+t_{ij} \le t_j$

- in the discrete interval $[a_j, b_j]$ if $b_i + t_{ij} \le b_j$
- in the discrete rectangle $[b_j - t_{ij}, b_i] \times [a_j, b_j]$ if $b_i + t_{ij} > b_j$.

The efficiency of this probability calculation depends on the width of the time intervals. Thus, a trip without predecessor must necessarily begin at time $t_i = a_i$, while the probability estimate establishes it as beginning at b_i as in the worst possible case. A prior reduction in the time intervals was therefore made by successively applying a longest path algorithm from the depot s towards the set of trips so as to reduce the upper bounds of the intervals, and then applying the same algorithm in reverse from the depot t towards the set of trips to reduce the lower bounds of the intervals. The Ford-Bellman algorithm was used for this. When the time intervals have been reduced all trips without predecessors and trips without successors have their respective time intervals reduced to a single point in time, while the other trips have reduced intervals $[INF_i, SUP_i]$. Note that this reduction in the time intervals has no effect on the number of intertrip arcs (the set I) in the network.

5. APPLICATION TO SCHOOL TRANSPORTATION

5.1 Description of the problems studied

The two types of branching were tested on eight school transportation problems. Each problem is labelled according to the number of trips, followed by AM or PM to indicate whether morning or afternoon transportation is being considered. AM and PM problems involve the same trips with time intervals of the same width initially although the intervals are distributed differently over time. School finishing times are spread out over a larger time period than starting times as indicated in Table 1. The number of possible intertrip arcs is therefore greater in the afternoon and the problems are very different. The effect of reducing the time intervals is not the same for the AM and PM problems (see Table 2).

Problem size	Number of schools	Duration of trips			Duration of intertrip arcs			Number of arcs		Total duration	
		Min	Max	Av.	Min	Max	Av.	AM	PM		
48	6	8	33	18.7	1	29	10.9	579	620	80	110
100	46	13	76	34.4	1	56	19.0	1137	2088	119	166
151	55	8	76	32.9	1	56	18.5	2620	4900	121	166
223	65	8	76	32.6	1	56	18.3	6273	10792	131	166

Table 1
Description of problems

Problem name	Interval width (b_i-a_i) of original problems			Interval width (b_i-a_i) after reduction		
	% at 0	Max	Av.	% at 0	Max	Av.
48 AM	15%	9	4.6	73%	7	1.0
48 PM				71%	9	1.5
100 AM	34%	22	7.1	86%	22	2.5
100 PM				61%	22	4.0
151 AM	31%	22	7.8	79%	22	3.5
151 AM				56%	22	5.1
223 AM	31%	22	7.8	79%	22	3.6
223 AM				53%	22	5.3

Table 2
Effect of reducing the time intervals

In addition to the eight original problems, forty variants were studied. For each problem, we solved a variant with a fixed schedule (for comparison) where each trip i must begin at time a_i : this variant is identified by adding "$t_i=a_i$" to the problem name (e.g. 48 AM $t_i=a_i$). For each problem, we also modified the width of the original time intervals by replacing $[a_i,b_i]$ by $[a_i,b_i+\Delta]$ with Δ taking values of 5, 10, 15 and 20 minutes; the variants of this type are identified by the original problem name followed by + Δ, (e.g. 48 AM + 20).

5.2 Aggregation of the network

The schools form the common origin and destination of several trips and the size of the network can be reduced by exploiting this specific structure. Most of the arcs of the network are intertrip arcs and their number can be reduced by using "school" nodes.

Consider the problem of homeward trips from the school in the afternoon. The nodes of the network include the depot exit and entrance nodes, the set of trips P and the set of schools E. To reduce the arc structure, some additional notation most be introduced.

P_e : the set of trips originating at the school $e \in E$; $\{P_e, e \in E\}$ forms a partition of the set of trips P;

$E_i = \{e \in E \mid (\forall j \in P_e),\ b_i + t_{ij} \leq a_j,\ i \in P\}$; a school e belongs to E_i if after completing trip i, there is a wait greater than or equal to zero before the start of <u>all</u> trips $j \in P_e$ starting from this school.

The set I of intertrip arcs $(i,j) \in P \times P$ satisfying relation (1) is replaced by three sets of arcs : I_1 (the set of trip-school arcs), I_2 (the set of school-trip arcs) and I_3 (the subset of intertrip arcs in I not replaced by I_1 or I_2).

$$I_1 = \bigcup_{i \in P} (\{i\} \times E_i)$$

$$I_2 = \bigcup_{e \in E} (\{e\} \times P_e)$$

$$I_3 = I - \bigcup_{i \in P} \bigcup_{e \in E_i} (\{i\} \times P_e)$$

The set of new network arcs includes the arcs from the depot to the schools $\{s\} \times E$, the arcs $I_1 \cup I_2 \cup I_3$ and the arcs returning the vehicles to the depot $P \times \{t\}$. This aggregation remains valid for all nodes in the branch-and bound tree. When the schedules are fixed ($a_i = b_i\ \forall i \in P$), $I_1 \cup I_2$ entirely replaces I and it can be shown that the number of arcs of the network decreases from $O(|P|^2)$ to $O(|P| * |E|)$. Although less significant when the intervals are large, this reduction remains generally advantageous.

Problem name	Original problem		Problem variants							
			Δ = 5		Δ = 10		Δ = 15		Δ = 20	
48 AM	675[1]	380[2]	765	557	869	713	972	864	1095	1033
48 PM	716	350	807	474	896	625	984	792	1095	947
100 AM	1337	1250	1732	1678	1993	1969	2293	2282	2705	2720
100 PM	2288	1653	2678	2109	2997	2560	3218	2919	3538	3322
151 AM	2922	2676	3874	3724	4613	4513	5410	5367	6370	6374
151 PM	5202	3331	6089	4430	6831	5590	7379	6571	8097	7529
223 AM	6719	5937	8726	8199	10477	10121	12269	12058	14176	14116
223 PM	11238	6852	13261	9462	14933	11987	16141	14093	17796	16341

[1] Number of arcs before aggregation
[2] Number of arcs after aggregation

Table 3
Effect of aggregation on the size of the network

The afternoon problem involves bringing school children back to their homes as soon as possible, while the morning problem requires them to be brought to school as late as possible. The network for the morning problem is therefore constructed in reverse, i.e. the time intervals are reversed :

$$[a_i, b_i] := [-b_i, -a_i] \qquad i \in P \qquad (28)$$

and the same procedure for the construction of the network is applied with the arcs in the opposite direction.

5.3 Tests on the original problems

Two tables of results are presented in this section : the first measures the effect of using probabilities rather than intertrip travel times as a secondary term in the objective function, while the second compares branching on flow variables with branching on time intervals.

The effect of the choice of objective function is presented using the time interval branching method (Table 4), but the results are similar using flow variable branching. Including the probability terms in the objective function significantly reduces the number of illegal routes (no subtours in the original problems) by favouring, for a given number of buses, the inclusion in intertrip arcs with a higher probability of feasibility. This means that a first feasible solution as well as the best feasible solution are found much more quickly. It also means that optimality is proved more quickly.

Table 5 demonstrates the unquestionable superiority of the time interval branching method over the flow variable branching method with the probabilities included in the objective function. In the first case, 6 out of 8 of the original problems were solved to optimality in a very short time as compared with optimal solutions produced by column generation [8] (not available for the problems with 223 trips), while only four problems were solved to optimality when branching on flow variables. In addition, for the problems which were not solved to optimality with the flow variable branching method, the best solution found was sometimes very far from optimal or from the best solution found using the time interval branching method.

For the eight original problems, using the time interval branching method, the best solution is found after exploring very few nodes and the maximum level in the tree is relatively small as compared with the results obtained when branching on flow variables. In the specific case of problem 223 AM, the first feasible solution is found at node 45 (level 44) while the optimal solution is found at node 47 (level 45); the exploration of the tree ended in this problem with the analysis of the bound calculated at each pending node of the tree. On the other hand, the best solution for problem 151 AM, found after exploring only 36 nodes (around 16 seconds), involves 94 vehicles; this solution is optimal but cannot be proved so within a reasonable time. In fact, after 390 seconds, around ten levels in the branch-and-bound tree remain to be explored, and all bounds suggest solutions of 93 vehicles, i.e. solutions which are infeasible. The secondary term is thus no longer effective here and the exploration through ten levels of all these infeasible combinations with 93 vehicles

Problem name	Network relaxation		First solution		Best solution		Branch-and-bound tree		Time (sec)
	Buses	Illegal routes	Buses	Nodes	Buses	Nodes	Nodes explored	Maximum level	CDC 173
48 AM	17	1[1]	17	3	17	3	3	2	1
		1[2]	17	2	17	2	2	1	1
48 PM	17	3	17	21	17	21	21	20	3
		0	17	1	17	1	1	0	1
100 AM	63	6	64	19	63	212	212	18	49
		1	63	2	63	2	2	1	3
100 PM	48	7	51	18	51	18	188	17	51
		4	51	7	51	7	53	7	17
151 AM	93	11	95	37	94	42	865	38	> 390
		5	95	24	94	36	893	32	> 390
151 PM	72	12	76	37	75	41	137	38	74
		6	75	13	75	13	120	12	71
223 AM	134	14	136	61	135	62	111	61	95
		11	136	45	135	47	47	45	44
223 PM	104	16	109	65	107	86	467	64	> 390
		10	107	17	107	17	323	16	> 390

[1] $Z = C * \sum_{i} x_{sj} + \sum_{ij} t_{ij} x_{ij}$ [2] $Z = C * \sum_{i} x_{sj} - \sum_{ij} (\ell n\ p_{ij})\ x_{ij}$

Table 4
Choice of the secondary term when optimizing
on the number of vehicles only (branching on time windows)

is practically speaking impossible to carry out. The same remarks apply to problem 223 PM. It should be noted that if such an enumeration is necessary, the tree can be dealt with more rapidly by transforming the general network problem in a maximum flow problem from s to t.

Finally, the execution times (on CDC 173) presented in this section and the next include time for the construction of the aggregated network, the reduction of the time intervals using a longest path algorithm twice, the evaluation of probabilities, the solution of the network relaxation and subsequent reoptimizations as well as time for the management of the branch-and-bound tree and input-output time.

5.4 Effect of time interval width

Table 6 shows results for the variants of the problems in which the intervals $[a_i, b_i]$ were gradually widened until the width was of the same order of magnitude as the duration of a trip in the case $\Delta = 20$. The variants are generally much more difficult to solve as they deviate more and more from the pure network problem of route construction with fixed schedules (i.e. with $t_i = a_i$).

Problem name	Network relaxation		First solution		Best solution		Optimal solution by column generation [8]		Branch-and-bound tree		Time (sec)
	Buses	Illegal routes	Buses	Nodes	Buses	Nodes	Buses	Nodes	Nodes explored	Maximum level	CDC 173
48 AM	17	1	18[1]	2	17	5	17	21	5[1]	2	1
			17[2]	2	17	2			2[2]	1	1
48 PM	17	0	17	1	17	1	17	16	1	0	1
			17	1	17	1			1	0	1
100 AM	63	1	64	2	63	3	63	26	3	2	3
			63	2	63	2			2	1	3
100 PM	48	4	53	7	51	64	51	59	772	41	> 193
			51	7	51	7			53	7	17
151 AM	93	5	95	6	95	6	94	94	523	56	> 193
			95	24	94	36			893	32	> 390
151 PM	72	6	80	13	77	244	75	376	393	80	> 193
			75	13	75	13			120	12	71
223 AM	134	11	137	11	135	156	not available		156	73	127
			136	45	135	47			47	45	44
223 PM	104	10	116	13	113	214	not available		227	60	> 243
			107	17	107	17			323	16	> 390

[1] branching on flow variables

[2] branching on time intervals

Table 5
Branching on flow variables vs branching on time intervals

Problem name	Network relaxation Buses	I.I.[1]	S.T.[2]	First solution Buses	Nodes	Best solution Buses	Nodes	Optimal solution by column generation [8] Buses	Time[4] (sec)	Branch-and-bound tree Nodes explored	Maximum level	Time (sec) CDC 173
48 AM $t_i=a_i$	20	0	0	20	1	20	1	20	4	1	0	1
48 AM	17	1	0	17(18)	2	17(17)	2	17	20	2	1	1
48 AM + 5	15	3	1	15(17)	12	15(15)	12	15	24	12	11	2
48 AM + 10	12	7	0	14(16)	33	14(15)	33	14	82	546	41	> 93
48 AM + 15	10	10	0	14(16)	51	13(14)	417	13	179	517	60	> 93
48 AM + 20	6	6	1	14(15)	91	13(14)	102	12	155	929	102	> 193
48 PM $t_i=a_i$	24	0	0	24	1	24	1	24	4	1	0	1
48 PM	17	0	0	17(17)	1	17(17)	1	17	16	1	0	1
48 PM + 5	14	4	0	14(15)	17	14(14)	17	14	67	17	16	3
48 PM + 10	13	5	0	14(14)	12	14(14)	12	13	124	2370	55	> 293
48 PM + 15	10	8	0	13(14)	46	13(13)	46	12	178	1780	81	> 293
48 PM + 20	9	8	1	13(13)	23	13(13)	23	12	367	1625	93	> 293
100 AM $t_i=a_i$	70	0	0	70	1	70	1	70	10	1	0	2
100 AM	63	1	0	63(64)	2	63(63)	2	63	26	2	1	3
100 AM + 5	56	6	0	59(60)	26	59(59)	26	59	52	1306	27	> 390
100 AM + 10	54	3	0	55(57)	20	55(55)	20	55	86	766	27	> 243
100 AM + 15	51	5	0	51(53)	22	51(51)	22	51	103	22	21	10
100 AM + 20	46	6	0	46(47)	19	46(47)	19	46	143	19	18	12
100 PM $t_i=a_i$	59	0	0	59	1	59	1	59	11	1	0	3
100 PM	48	4	0	51(53)	7	51(51)	25	51	59	53	7	17
100 PM + 5	42	2	0	47(47)	25	47(47)	175	46	73	1199	34	> 390
100 PM + 10	41	3	0	45(46)	22	44(45)	175	44	232	211	26	> 93
100 PM + 15	37	8	0	41(42)	26	40(41)	115	40	405	164	28	> 93
100 PM + 20	37	9	0	37(40)	18	37(40)	18	37	481	18	17	16
151 AM $t_i=a_i$	108	0	0	108	1	108	1	108	21	1	0	4
151 AM	93	5	0	95(93)	24	94(95)	36	94	94	893	32	> 390
151 AM + 5	82	13	0	89(90)	49	89(89)	49	88	210	234	48	> 144
151 AM + 10	77	13	0	84(85)	55	84(85)	55	82	382	621	59	> 390
151 AM + 15	73	11	0	75(79)	59	75(78)	59	74	511	535	59	> 390
151 AM + 20	67	13	0	71(71)	72	71(71)	72	-	-	376	78	> 390
151 PM $t_i=a_i$	92	0	0	92	1	92	1	92	32	1	0	5
151 PM	72	6	0	75(80)	13	75(77)	13	75	376	120	12	71
151 PM + 5	62	5	1	71(72)	37	70(72)	63	70	770	539	41	> 390
151 PM + 10	58	7	0	68(69)	48	67(68)	56	66	692	503	52	> 390
151 PM + 15	53	15	0	60(65)	56	59(64)	211	58	1146	318	55	> 390
151 PM + 20	53	15	0	56(61)	54	55(61)	59	-	-	236	55	> 390
223 AM $t_i=a_i$	155	0	0	155	1	155	1	-	-	1	0	6
223 AM	134	11	0	136(137)	45	135(135)	47	-	-	47	45	44
223 AM + 5	118	15	0	127(129)	64	126(127)	70	-	-	141	63	> 193
223 AM + 10	112	19	0	119(123)	82	119(121)	82	-	-	117	81	> 193
223 AM + 15	108	15	0	109(113)	87	108(112)	26	-	-	126	88	253
223 AM + 20	101	18	0	101(106)	75	101(106)	76	-	-	75	76	216
223 PM $t_i=a_i$	132	0	0	132		132	1	-	-	1	0	9
223 PM	104	10	0	107(116)	17	107(113)	17	-	-	323	16	> 390
223 PM + 5	85	10	0	99(105)	64	98(103)	71	-	-	115	66	> 193
223 PM + 10	80	9	1	94(99)	77	94(107)	77	-	-	81	76	> 193
223 PM + 15	75	22	0	85(93)	81	84(92)	103	-	-	154	80	> 193
223 PM + 20	-	-	-	-	-	-	-	-	-	-	-	-

[1] Number of illegal routes in the network relaxation.
[2] Number of subtours in the network relaxation.
[3] Number of buses for the time interval branching method with number for variable branching in parentheses.
[4] Computer time in seconds on CDC 173.

Table 6
Effect of width of the time intervals

The table shows the results using the time interval branching method with the results using branching on flow variables given in parentheses. Note again the superiority of the time interval branching results, even though few problems were solved to optimality.

Note also the appearance of subtours (of type Case 1 only), the increasing gap between the solution of the network relaxation and the optimal solution, and the rapid increase in the size of the tree. However certain problems with $\Delta = 15$ and $\Delta = 20$ were easily solved. This can be explained as follows : in school transportation, routes are usually made up of one, two or three trips. Routes of one or two trips are always feasible, while routes with three trips are not necessarily so. But when the intervals become sufficiently wide without generating subtours, these same routes with three trips become feasible.

Finally when both the number of vehicles and the travel times are optimized, the flow variable branching method was unable to solve optimally even the 48 AM problem in 150 seconds of computer time; the best solution was 25 minutes of travel time away from the optimum (gap of 13%). On the other hand, the two problems 48 AM and 48 PM are easily solved in 1.6 seconds (11 nodes) and 2.0 seconds (19 nodes) respectively using the time interval branching method. For the other problems, the method sometimes gives good solutions but never completes the exploration of the tree in a reasonable time. By way of comparison, the solution times for the column generation method are given for the simultaneous optimization of the number of vehicles and travel times [8], but these times are roughly the same if only the number of vehicles is optimized.

6. CONCLUSION

Relaxing the scheduling constraints to solve a network problem retains the integrality of the solution but the lower bound provided by this relaxation generally deteriorates as the number of trips or the width of the time intervals increases. The introduction of a secondary term made up of probabilities rather than intertrip travel times significantly reduces the number of illegal routes and enables a feasible solution to be found more rapidly in the branch-and-bound tree. Although this problem is a generalization of the m-travelling salesman problem, analysis of the numerical results demonstrates the superiority of the time interval branching method over the flow variables branching method used by Carpaneto and Toth [5] on the asymmetric travelling salesman problem.

The network relaxation method with branching on the time intervals can solve large problems. However, it is limited to problems with relatively narrow time windows. This method is suitable for applications with small time windows where a reduction in the number of vehicles by more than 10% is not to be expected as the schedules of the trips cannot be greatly modified.

REFERENCES

[1] Bellmore, Malone, Pathology of travelling salesman subtour elimination algorithm. Operations Research 19 (1971) 278-307.

[2] Bodin, Golden, Assad, Ball, The State of the Art in the Routing and Scheduling of Vehicules and Crews. (1981). Document available at National Technical Information Service, Springfield, VA 22161 U.S.A.

[3] Bratley, Florian, Robillard, Scheduling with earliest start and due date constraints on multiple machines. Naval Res. Logistics Quart. 22 (1975) 165-173.

[4] Christofides, Mingozzi, Toth, State-space relaxation procedures for the computation of bound to routing problems. Network 11 (1981) 145-164.

[5] Carpaneto, Toth, Some new branching and bounding criteria for the asymmetric travelling salesman problem. Management Science 26 (1980) 736-743.

[6] Desrosiers, Ferland, Rousseau, Lapalme and Chapleau, TRANSCOL : A Multi-Period School Bus Routing and Scheduling System. Publ. #164, Centre de recherche sur les transports, Université de Montréal (1980)

[7] Desrosiers, Pelletier, Soumis, Plus court chemin avec contraintes d'horaires. (1983). To appear in RAIRO.

[8] Desrosiers, Soumis, Desrochers, Routing with Time Windows by Column Generation. Publication #278, Centre de recherche sur les transports, Université de Montréal (1983).

[9] Fisher, Lagrangian relaxation method for integer programming problems. Management Science 27 (1981) 1-18.

[10] Geoffrion, Lagrange relaxation for integer programming. Math. Prog. Study 2 (1974) 82-114.

[11] Gertsbakh, Stern, Minimal resources for fixed and variable job schedules, Operations Research 26, 1, (1978).

[12] Levin, Scheduling and fleet routing models for transportation systems, Transportation Science 5, 3 (1971).

[13] Miller, Tucker, Zemlin, Integer programming formulation of travelling salesman problems. Journal of the Association of Computing Machinery, 6 (1960) 326-332.

[14] Orloff, Route constrained fleet scheduling. Transportation Science, 10 (1976) 149-168.

[15] Soumis, Desrosiers, Locomotive Maintenance Scheduling and Train Assignment. Publication #239, Centre de recherche sur les transports, Université de Montréal (1983).

[16] Soumis, Desrosiers, Desrochers, Optimal Routing and Scheduling for Public Buses with Flexibility in the Time Table. Publication #285, Centre de recherche sur les transports, Université de Montréal (1983).

[17] Soumis, Ferland, Rousseau, A model for large scale aircraft routing problems. Transportation Research B 14B, 1 (1980).

[18] Swersey, Ballard, Scheduling School Buses. Published at Yale School of Organization and Management and presented at TIMS/ORSA Conference, San Diego (1982).